ПАРАЛЛЕЛЬНОЕ ШИФРОВАНИЕ ДАННЫХ АЛГОРИТМОМ RSA

А.А. Неретин

Научный руководитель: доцент, к.ф-м.н. О.Л. Крицкий

Национальный исследовательский Томский политехнический университет,

Россия, г. Томск, пр. Ленина, 30, 634050

E-mail: drdmx@yandex.ru

PARALLEL DATA ENCRYPTION WITH RSA ALGORITHM

A.A. Neretin

Scientific Supervisor: PhD, Associate Prof. O.L. Kritski

Tomsk Polytechnic University, Russia, Tomsk, Lenin str., 30, 634050

E-mail: drdmx@yandex.ru

Abstract: *In this paper a parallel RSA algorithm with preliminary shuffling of source text was presented. Dependence of an encryption speed on the number of encryption nodes has been analysed, The proposed algorithm was implemented on C# language.*

Наиболее актуальной темой в XXI веке – веке информационных технологий - является работа с информацией, а главным приоритетом становится ее защита. Суть работы заключается в защите передаваемых данных и увеличении скорости обработки посредством нескольких этапов шифрования. Первый этап включает в себя предварительное перемешивание информации с использованием SHUFFLE алгоритма [1]. Второй этап – параллельное шифрование алгоритмом RSA на *n* узлах, где под узлом понимается персональный компьютер, входящий в сеть «Отправитель-Получатель». Разработанная нами схема представлена на рис. 1.

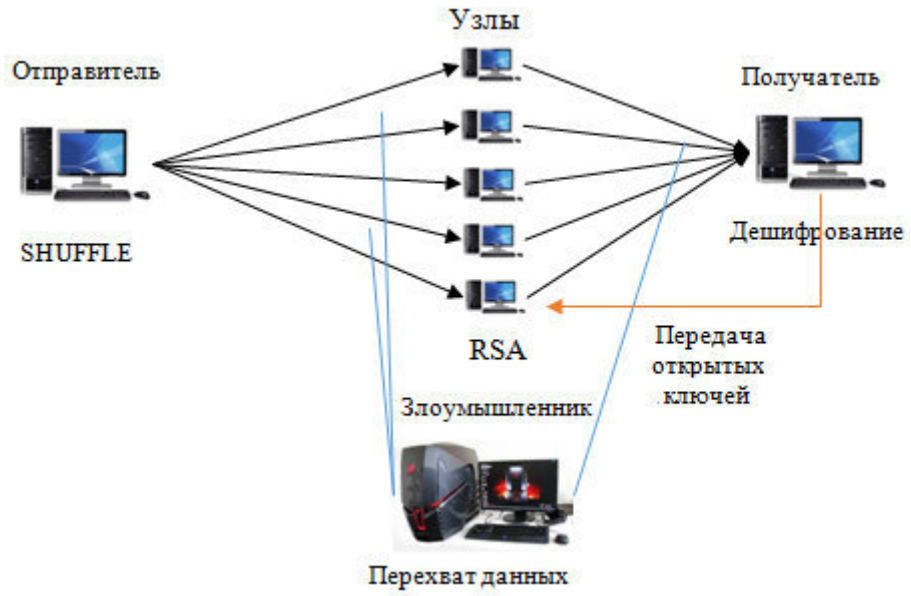

Рис. 1. Схема параллельного шифрования данных с возможностью перехвата информации

Так как алгоритм RSA относится к типу асимметричных, то скорость его работы намного ниже, чем у симметричных, но его криптоустойчивость кратно выше [2]. Параллельное шифрование частей текста на нескольких узлах значительно повышает скорость шифрации, а также повышает защищенность передаваемых данных, ведь если злоумышленник получит даже несколько частей текста на одном из двух этапов, то не сможет получить цельного текста, несущего смысл для получателя.

Основная идея в реализации данного алгоритма заключается в перемешивании изначального текста при помощи SHUFFLE алгоритма, разделении результата на *n* равных частей (дабы распределить нагрузку между узлами) и отправка текстов на компьютеры, входящих в цепь параллельного шифрования. Для шифрации получатель генерирует *n* парных ключей (один для шифрования, а второй для дешифрования) и производит их передачу на узлы. После получения части текста каждый узловой компьютер производит шифр-текст и передаёт его в точку приёма информации, для дальнейшей дешифрации и сбора цельного текста [3].

При параллельной шифрации данных скорость зависит от нескольких факторов:

1. длина текста;
2. используемый язык (английский, русский);
3. длина ключа;
4. количество узлов.

Результаты измерения скорости шифрования текстов при изменении различных факторов представлены в табл. 1. Измерения скорости производились на нескольких текстах равной длины, но разного содержания (литературный, технический, научный).

Таблица 1

Скорость шифрования при различном количестве узлов, с

	Без узлов	2 узла	4 узлов	8 узлов	16 узлов
Текст на английском длиной 1600	0,042	0,029	0,022	0,0206	0,0201
Текст на русском языке длиной 1400	0,040	0,030	0,027	0,0263	0,0231

Английский текст длиной 1600 слов имеет 8760 символов (35 040 байт), русский текст длиной 1400 слов, в свою очередь имеет 8674 символа (34696 байт) в кодировке UTF-32.

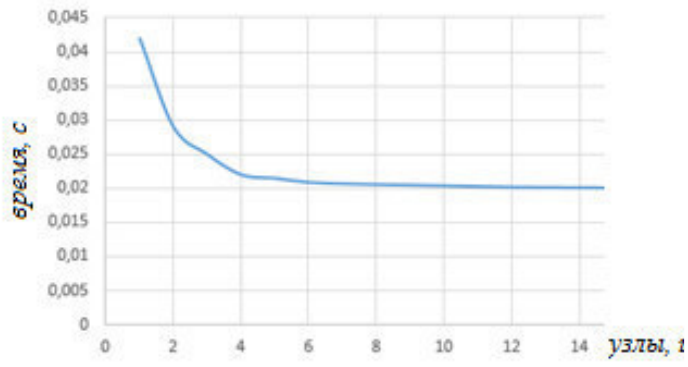

Рис. 2. Изменение скорости шифрования в зависимости от количества узлов

Криптостойкость алгоритма RSA основывается на сложности подсчёта обратной функции к функции шифрования. Для её расчёта нужно вычислить функцию Эйлера от открытого ключа *n*. Эта задача сводится к разложению (факторизации) данного числа на простые множители. Самый эффективный алгоритм выполнения факторизации на данный момент – общий метод решета числового поля

(РЧП). Криптостойкость алгоритма оценивается временем, которое требуется для факторизации k-битного числа методом РЧП, которое можно оценить как:

$$\exp(c+o(1))k^{1/3}\log^{1/3}k,$$

где $c<2$ – коэффициент который, зависит от вычислительной мощности компьютера.

В связи с постоянным ростом вычислительных мощностей для шифрации использовался ключ длиной 2048 бит, что равно (2^{2047} – 617)-значному числу. На сегодняшний день ключ длиной 2048 бит является криптоустойчивым, для его взлома потребуется $2{,}042\cdot10^8$ лет. К настоящему времени явно был взломан лишь ключ длиной 700 бит, при этом процесс занял 11 месяцев совместной работы 300-400 компьютеров (2007 год). Реализованная схема так же повышает криптостойкость в n раз и исключает полный перехват информации, что почти на 100% защищает пользователя, так же если будет произведена подмена пакетов данных, то получатель сможет легко найти брешь в защите, ведь он будет знать номер узла, в котором произошла подмена данных.

Анализ результатов, представленных в табл. 1, показывает, что оптимальным количеством узлов для эффективного ускорения шифрования по данному методу является 8-16 узлов, использование которых ускоряют вычисление на 42,5-52% относительно скорости шифрования на персональном компьютере. Дальнейшее увеличение узлов не придает весомого ускорения и может быть использовано лишь для повышения криптостойкости. Защита будет повышаться прямо пропорционально увеличению количества узлов, ведь будут появляться новые парные ключи, которые так же нужно будет взламывать.

Алгоритм SHUFFLE намного быстрее, и выполняется за 0,0001 с для текста почти любой длины (рассматривался текст длиной 30 000 символов и 3 циклами перемешивания), при уменьшении или увеличении его размера на скорость это никак не влияет, влияет только разрядность шины данных микропроцессора.

Реализация схемы параллельного шифрования выполнена на языке C# в среде разработки Visual Studio. Для сопоставления буквам и символам числового эквивалента использовались значения кодировки UTF-32, после шифрования выполнялось обратное преобразование в Base64String, данная функция при конвертации использует сопоставление чисел с символами в кодировке ASCII. Это позволяет производить однозначное преобразование для шифрации и дешифрации.

Данные исследования могут быть полезны в экономической сфере. Предложенный алгоритм имеет применение при передаче новостных текстов, секретных документов, предназначенных для закрытого круга лиц [3]. Из-за отсутствия эффективного метода для факторизации чисел длиной большей, чем 230 разрядов, сгенерированные единожды ключи могут использоваться на протяжении нескольких лет, а может и десятилетий. Также в любой момент можно произвести замену ключей, включенных в систему.

СПИСОК ЛИТЕРАТУРЫ

1. Margonda J. Perfect Shuffle Algorithm for Cripthography. ARPN Journal of Engineering and Applied Sciences. – 2014. – V. 9 (12), p. 2384-2386.
2. Петров А.А. Компьютерная безопасность. Криптографические методы защиты. – М.: ДМК, 2000. – 449 с.
3. Яковлев А.В. Криптографическая защита информации: учебное пособие/ Яковлев А.В., Безбогов А.А., Родин В.В., Шамкин В.Н. – Тамбов: Изд-во Тамб. гос. техн. ун-та, 2006. – 140 с.